PHP - Les cookies et les en-têtes HTTP

Juin 2014

- Les en-têtes HTTP
- Récupérer les en-têtes de la requête
- Créer le cookie
- Quelques précisions sur les cookies
- Exemple d'utilisation de cookies
- Récupérer les valeurs d'un cookie
- Supprimer un cookie
- Quelques réalisations avec les cookies

Les en-têtes HTTP

Lors de chaque échange par le protocole HTTP entre votre navigateur et le serveur, des données dîtes d'en-têtes contenant des informations sur les données à envoyer (dans le cas d'une requête) ou envoyées (dans le cas d'une réponse). Les informations en question, généralement sur une page web ou une image, suivent ces en-têtes. Les en-têtes HTTP permettent aussi d'effectuer des actions sur le navigateur comme le transfert de cookie ou bien une redirection vers une autre page.

Ces en-têtes sont les premières informations envoyées au navigateur (pour une réponse) ou au serveur (dans le cas d'une requête), elles se présentent sous la forme :

```
en-tête: valeur
```

La syntaxe doit être rigoureusement respectée, c'est-à-dire qu'aucun espace ne doit figurer entre le nom de l'en-tête et les deux points (:). Un espace doit par contre figurer après celui-ci !

PHP fournit une fonction permettant d'envoyer très simplement des en-têtes HTTP manuellement du serveur au navigateur (il s'agit alors d'une réponse HTTP. La syntaxe de cette fonction est la suivante :

```
booléen header(chaîne en-tête HTTP)
```

Etant donnée que les en-têtes HTTP sont les premières informations envoyées, la fonction *header()* doit être utilisée avant tout envoi de données HTML au navigateur (le script qui la

contient doit donc être placé avant la balise <HTML> et avant toute fonction *echo(),print ou printf()*)

Voici quelques utilisations possibles de la fonction *header()* :

- pour rediriger le navigateur vers une nouvelle page :

```
<?
header("location: http://www.commentcamarche.net/");
?>
```

- Pour envoyer au navigateur une image créé à la volée (pour faire un compteur de statistiques ou bien un histogramme dynamique par exemple) :

```
<?
header("Content-Type: image/gif");
// code générant l'image
imagegif($image); // envoi de l'image au navigateur
?>
```

Récupérer les en-têtes de la requête

Alors que la fonction *header()* permet d'envoyer des en-têtes HTTP au navigateur, PHP fournit une seconde fonction permettant de récupérer dans un tableau l'ensemble des en-têtes HTTP envoyées par le navigateur. Voici la syntaxe de cette fonction :

```
Tableau getallheaders();
```

Le tableau retourné par la fonction contient les en-têtes indexés par leur nom. Voici un script permettant par exemple de récupérer des en-têtes particuliers.

```
<?
$entetes = getallheaders;
echo $entetes["location"];
?>
```

Créer le cookie

Un cookie) est un petit fichier texte stocké sur le disque dur du client, permettant de mémoriser individuellement des préférences pour chaque utilisateur.

Pour créer un cookie sur une machine appelant un de vos scripts PHP, il faut utiliser la fonction *setcookie()* dans celui-ci. La fonction *SetCookie()* permet d'envoyer les informations relatives aux cookies dans les en-têtes HTTP
Voici la syntaxe de la fonction *setcookie* :

```
booléen setcookie(chaîne NomDuCookie,
 chaîne Valeur,
 entier expiration,
 chaîne chemin,
 chaîne domaine,
 entier securisé);
```

L'argument *NomDuCookie* est celui qui vous permettra de faire référence à un cookie spécifique stocké sur le disque des utilisateurs. A chaque cookie correspond un nom auquel une valeur est attribuée.

Les arguments suivants servent à déterminer les conditions de validité du cookie :

- **expiration**: détermine le moment à partir duquel le cookie sera effacé du disque du client. Elle doit être passée sous forme d'un entier indiquant le nombre de secondes à compter du 1er janvier 1970 à partir desquelles le cookies n'est plus valide. Il est généralement d'usage d'utiliser la fonction time() qui retourne le nombre de secondes de cette date à maintenant et d'y ajouter le nombre de secondes de validités que l'on désire. Pour un cookie valide pendant un an ce sera *time()+31536000*. Dans certains cas la fonction mktime() peut s'avérer elle aussi très pratique
- **chemin** désigne le répertoire à partir de la racine de votre domaine pour lequel votre cookie est valide, c'est-à-dire que si vous ne voulez utiliser le cookie que dans des scripts stockÉs dans le répertoire */commerce* il faudra définir le chemin */commerce/*
- **domaine** il s'agit du domaine de votre serveur, celui-ci doit obligatoirement contenir 2 points (par exemple *www.commentcamarche.net*). Par défaut (si vous omettez ce paramètre) le domaine qui a créé le cookie (donc le votre) sera automatiquement inséré
- **sécurisé** lorsqu'il est mis à 1 indique que le cookie ne sera transmis que si la ligne est sécurisée (par SSL ou S-HTTP

Quelques précisions sur les cookies

- La fonction *setcookie()* doit être utilisée avant tout envoi de données HTML vers le navigateur, même si ces données sont envoyées avec echo, print ou équivalent (le script qui la contient doit donc être placé avant la balise <HTML> et avant toute fonction *echo(),print ou printf()*).
- Le cookie n'est pas visible avant le prochain chargement de page.
- Avec PHP3 si vous envoyez plusieurs cookies de suite, les appels seront traités en ordre inverse, alors qu'avec PHP4 ils seront traités dans l'ordre.
- Il est possible d'utiliser des tableaux dans un cookie. Autant de cookies que d'éléments du tableau seront alors envoyés, mais tout se fait de façon transparente, puisque à la lecture un tableau sera créé. Il est quand même préférable d'utiliser les fonctions implode et explode pour envoyer ainsi qu'un seul cookie.
- Il faut savoir que certains navigateurs ne traitent pas bien les cookies
    - *Microsoft Internet Explorer 4* avec le *Service Pack 1* ne traite pas correctement les cookies qui ont le paramètre chemin défini.

- Inversement *Netscape Communicator 4.05* et *Microsoft Internet Explorer 3.x* ne traitent pas correctement les cookies qui n'ont pas les paramètres chemin et expiration définis.

Exemple d'utilisation de cookies

Voilà donc un script permettant de savoir si un visiteur est déjà venu sur le site pendant le mois :

```
<?php
setcookie(
"Visites",
"Oui",
time()+2592000,
"/",
".commentcamarche.net",0);
?>
```

Quelques autres exemples :

```
<?php
// Envoi d'un cookie qui disparaîtra après la fermeture du navigateur
SetCookie("CcmUserSessionCookie","$login:$pass");

// Envoi d'un cookie qui restera présent 24 heures
SetCookie("CcmDejaVisite","1",time()+3600*24,"/",".phpfrance.com",0);

// Envoi d'un cookie qui s'effacera le 1er janvier 2001
SetCookie("CcmAn2000","1",mktime(0,0,0,1,1,2001),"/",".phpfrance.com",0);

?>
```

Récupérer les valeurs d'un cookie

Lorsqu'un cookie pour votre domaine est présent sur le disque du client, si celui-ci visite une de vos pages PHP dont le chemin est inclut dans le paramètre chemin (si le chemin est /le cookie est valide pour toutes les pages de votre site) le cookie sera automatiquement passé en paramètre à votre script et sa valeur sera directement accessible dans la variable *$NomDuCookie*.

Voici le même script que précédemment permettant cette fois ci de compter le nombre de visite de la page par le visiteur :

```
<?php
$Visites++;

setcookie(
 "Visites",
 $Visites,
 time()+2592000,
 "/",
 ".commentcamarche.net",0);
?>
```

A chaque appel de la page, le script récupère la valeur du cookie nommé *Visites*, l'incrémente, puis envoie le cookie dont la valeur a été incrémentée.

Supprimer un cookie

Il peut éventuellement être utile de proposer à l'utilisateur de supprimer certains de ses cookies, ou bien de le faire vous même de façon transparente.

Pour ceci il suffit de renvoyer le cookie grâce à la fonction *setcookie()* en spécifiant simplement l'argument *NomDuCookie* :

```
<?
setcookie("Visites");
?>
```

Une autre méthode consiste à envoyer un cookie dont la date d'expiration est passée :

```
<?
setcookie("Visites","",0,"/","",0);
?>
```

Quelques réalisations avec les cookies

Lors d'une inscription à CCM, un mot de passe et un identifiant vous sont demandés. Après confirmation de votre inscription en renvoyant simplement le mail qui vous a été automatiquement envoyé, vous pouvez utiliser votre login et votre mot de passe dans la section s'identifier, activant un cookie sur votre disque dur, contenant un numero d'identification ainsi qu'un numéro aléatoire permettant au site de vous reconnaître dans certaines sections en comparant ces valeurs avec celles stockées dans une entrée de la base de donnée. Cela fait apparaître de nouvelles options aux membres CCM sans gêner les non-membres !

Ce document intitulé « PHP - Les cookies et les en-têtes HTTP » issu de **CommentCaMarche** (www.commentcamarche.net) est mis à disposition sous les termes de la licence Creative Commons. Vous pouvez copier, modifier des copies de cette page, dans les conditions fixées par la licence, tant que cette note apparaît clairement.